

JUN 16 2005

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400PATENT APPLICATION
ATTORNEY DOCKET NO. 200301685-1IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): David J. LINDNER

Confirmation No.: 7765

Application No.: 09/611,920

Examiner: R. Singh

Filing Date: 07/07/2000

Group Art Unit: 2176

Title: METHOD AND APPARATUS FOR PROVIDING ENHANCED ACCESS TO A LIGHTWEIGHT
DIRECTORY ACCESS PROTOCOL (LDAP) DIRECTORY SERVERMail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450RECEIVED
OIPE/IAP

JUN 20 2005

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed
on 04/22/2005.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) \$500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

() (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)
for the total number of months checked below:

| | |
|------------------|-----------|
| () one month | \$120.00 |
| () two months | \$450.00 |
| () three months | \$1020.00 |
| () four months | \$1590.00 |

() The extension fee has already been filed in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is
being made to provide for the possibility that applicant has inadvertently overlooked the need
for a petition and fee for extension of time.Please charge to Deposit Account 08-2025 the sum of \$500.00. At any time during the
pendency of this application, please charge any fees required or credit any over payment to Deposit
Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account
08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of
Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.() I hereby certify that this correspondence is being
deposited with the United States Postal Service as
first class mail in an envelope addressed to:
Commissioner for Patents, Alexandria, VA
22313-1450. Date of Deposit: _____
OR(X) I hereby certify that this paper is being transmitted
to the Patent and Trademark Office facsimile
number (703) 872-9306 on 06/16/2005

Number of pages: 32

Typed Name: Mark E. Scott

Signature: _____

Rev 12/04 (Apbriaf)

Respectfully submitted,

David J. LINDNER

By: _____

Mark E. Scott

Attorney/Agent for Applicant(s)
Reg. No. 43,100

Date: 06/16/2005

Telephone No.: (713) 238-8000

RECEIVED
CENTRAL FAX CENTER

JUN 16 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | | | |
|-------------|---------------------------|---|-------------------|-------------|
| Appellant: | David J. LINDNER | § | Confirmation No.: | 7765 |
| | | § | | |
| Serial No.: | 09/611,920 | § | Group Art Unit: | 2176 |
| | | § | | |
| Filed: | 07/07/2000 | § | Examiner: | R. Singh |
| | | § | | |
| For: | Method And Apparatus | § | Docket No.: | 200301685-1 |
| | For Providing Enhanced | § | | |
| | Access To A Lightweight | § | | |
| | Directory Access Protocol | § | | |
| | (LDAP) Directory Server | § | | |

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: June 16, 2005

Sir:

Appellant hereby submits this Appeal Brief in connection with the above-identified application. A Notice of Appeal was filed via facsimile on April 22, 2005.

06/17/2005 MBIZUNES 00000065 082025 09611920
01 FC:1402 500.00 DA

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

TABLE OF CONTENTS

| | | |
|-------|--|----|
| I. | REAL PARTY IN INTEREST | 3 |
| II. | RELATED APPEALS AND INTERFERENCES | 4 |
| III. | STATUS OF THE CLAIMS | 5 |
| IV. | STATUS OF THE AMENDMENTS | 6 |
| V. | SUMMARY OF THE CLAIMED SUBJECT MATTER | 7 |
| VI. | GROUND OF REJECTION TO BE REVIEWED ON APPEAL | 11 |
| VII. | ARGUMENT | 12 |
| | A. Claims 1-20 | 12 |
| | B. Claims 21, 23-27, 29-33 and 35-44 | 15 |
| | C. Non-Statutory Subject Matter Rejections | 18 |
| VIII. | CONCLUSION | 21 |
| IX. | CLAIMS APPENDIX | 22 |

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas, through its merger with Compaq Computer Corporation (CCC) that owned Compaq Information Technologies Group, L.P. (CITG). The assignment from the CCC to CITG was recorded on November 3, 2004, at Reel/Frame 015332/0477. The Change of Name document from CITG to HPDC was recorded on November 3, 2004, at Reel/Frame 015332/0427.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals or interferences.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

III. STATUS OF THE CLAIMS

Originally filed claims: 1-20.
Added claims: 21-44.
Cancelled claims: 22, 28 and 34.
Presently pending claims: 1-21, 23-27, 29-33 and 35-44.
Presently appealed claims: 1-21, 23-27, 29-33 and 35-44.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

IV. STATUS OF THE AMENDMENTS

There were no after-final amendments.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The Specification is directed to a method and apparatus for providing enhanced access to a directory server.¹ Some of the illustrative embodiments are a method for accessing a directory server, the method comprising simultaneously maintaining a first plurality of connections between the directory server and a caching daemon,² determining if an application is requesting information from the directory server,³ determining if the requested information is stored in the caching daemon in response to determining that the application has requested information,⁴ and sending the requested information to the application.⁵

Other illustrative embodiments are apparatus comprising a directory server for storing information,⁶ and a caching daemon maintaining a plurality of connections to the directory server.⁷ The caching daemon is configured to determine if an application is requesting information from the directory server,⁸ determine if the requested information is stored within the caching daemon,⁹ and send the requested information to the application.¹⁰

Other illustrative embodiments, *claimed in a means-plus-function format*, are an apparatus for accessing a directory server comprising a means for establishing a first plurality of simultaneously running connections¹¹ between the directory server and a caching daemon, means for determining if an application is requesting information from the directory server,¹² means for determining if the

¹ Specification Title (as amended August 10, 2004).

² Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

³ Specification Page 11, lines 15-16.

⁴ Specification Page 11, lines 16-20.

⁵ Specification Page 11, lines 22-23; Page 12, lines 6-8.

⁶ Specification Page 7, lines 12-21.

⁷ Specification Page 8, lines 2-7.

⁸ Specification Page 10, lines 7-10.

⁹ *Id.*

¹⁰ Specification Page 10, lines 10-12, lines 18-19.

¹¹ Specification Page 8, lines 2-7; Page 9, lines 1-5.

¹² Specification Page 10, lines 7-10.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

requested information is stored in the caching daemon in response to determining that the application has requested information,¹³ and means for sending the requested information to the application.¹⁴

Other illustrative embodiments are a caching daemon comprising a data cache adapted to store a subset of information from a directory server,¹⁵ and a controller.¹⁶ The controller is adapted to establish and maintain a first plurality of connections to the directory server,¹⁷ determine if an application is requesting information from the directory server,¹⁸ determine if the requested information is stored in the data cache,¹⁹ and send the requested information to the application.²⁰

Other illustrative embodiments are a method for accessing a directory server comprising simultaneously maintaining a plurality of connections between the directory server and a lightweight directory access protocol ("LDAP") caching daemon,²¹ determining if an application is requesting information from the directory server,²² determining if the requested information is stored in the LDAP caching daemon in response to determining that the application has requested information,²³ and sending the requested information to the application.²⁴

Other illustrative embodiments are an apparatus comprising a directory server for storing information,²⁵ and a lightweight directory access protocol

¹³ *Id.*

¹⁴ Specification Page 10, lines 10-12, lines 18-19.

¹⁵ Specification Page 10, lines 5-7.

¹⁶ Specification Page 10, lines 7-10.

¹⁷ Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

¹⁸ Specification Page 10, lines 7-10.

¹⁹ *Id.*

²⁰ Specification Page 10, lines 10-19.

²¹ Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

²² Specification Page 11, lines 15-18.

²³ Specification Page 11, lines 16-20.

²⁴ Specification Page 11, lines 22-23; Page 12, lines 6-8.

²⁵ Specification Page 7, lines 12-21.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

("LDAP") caching daemon.²⁶ The LDAP caching daemon is configured to maintain a plurality of connections to the directory server,²⁷ determine if an application is requesting information from the directory server,²⁸ determine if the requested information is stored within the LDAP caching daemon,²⁹ and send the requested information to the application.³⁰

Other illustrative embodiments are a lightweight directory access protocol ("LDAP") caching daemon comprising a data cache that stores a subset of information from a directory server,³¹ and a controller configured to establish and maintain a first plurality of connections to the directory server,³² determine if an application is requesting information from the directory server,³³ determine if the requested information is stored in the data cache,³⁴ and send the requested information to the application.³⁵

Other illustrative embodiments are a method for accessing a directory server comprising continuously maintaining a plurality of connections between the directory server and a lightweight directory access protocol caching ("LDAP") caching daemon,³⁶ receiving from an application a request for information from the directory server,³⁷ using a connection of the plurality of connections to retrieve

²⁶ Specification Page 8, lines 2-7.

²⁷ Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

²⁸ Specification Page 10, lines 7-10.

²⁹ *Id.*

³⁰ Specification Page 10, lines 10-12, lines 13-19.

³¹ Specification Page 10, lines 5-7.

³² Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

³³ Specification Page 10, lines 7-10.

³⁴ *Id.*

³⁵ Specification Page 10, lines 10-19.

³⁶ Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

³⁷ Specification Page 11, lines 15-16.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

the requested information from the directory server,³⁸ and sending the requested information to the application.³⁹

Finally, other illustrative embodiments are a lightweight directory access protocol ("LDAP") caching daemon comprising a data cache that stores a subset of information from a directory server,⁴⁰ and a controller configured to establish and maintain a plurality of connections to a directory server,⁴¹ receive from an application a request for information from the directory server,⁴² use a connection of the plurality of connections to retrieve the requested information from the directory server,⁴³ and send the requested information to the application.⁴⁴

³⁸ Specification Page 10, lines 14-16.

³⁹ Specification Page 11, lines 22-23; Page 12, lines 6-8.

⁴⁰ Specification Page 10, lines 5-7.

⁴¹ Specification Page 8, lines 4-7; Page 9, lines 1-5; Page 11, lines 11-13.

⁴² Specification Page 10, lines 7-10.

⁴³ Specification Page 10, lines 14-16.

⁴⁴ Specification Page 10, lines 10-19.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-21, 23-27, 29-33 and 35-44 are unpatentable over an article title "Using LDAP Directory Caches," authored by Sophie Cluet, Olga Kapitskaia, and Divesh Srivastava (hereinafter the Cluet article), in view of Luotonen (U.S. Pat. No. 5,864,852).

Whether claims 16-20, 33, 34-38 and 42-44 are directed to statutory subject matter.

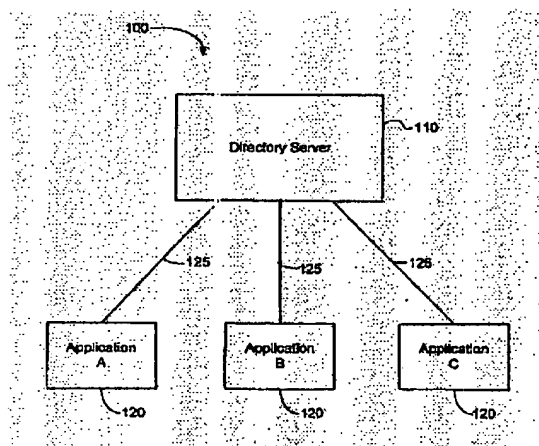
Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

VII. ARGUMENT

A. Claims 1-20

Claims 1-20 stand rejected as allegedly obvious over the Cluet article in view of Luotonen. Claim 1 is representative of this group of claims. This grouping is for purposes of this appeal only, and should not be construed to mean the patentability of any of the claims may be determined, in later actions before a court, based on the grouping. Rather, the presumption of 35 U.S.C. § 282 shall apply to each claim individually.

Appellant's Background section and corresponding related-art Figure 1 (reproduced immediately below) note a difficulty in the related art.



With regard to Figure 1, Appellant's Background section states:

Turning now to the drawings, and specifically referring to Figure 1, a system 100 for providing LDAP directory server access to a plurality of client server applications is shown in accordance with the prior art. Typically, when a client server application 120 desires to access data from a directory server 110, the application 120 establishes a direct connection 125 to the directory server 110 through a binding operation. ... Upon successfully establishing the connection 125 with the directory server 110, the application 120 then retrieves the desired data from the directory server 110 by performing a search operation. After retrieving this desired data, the client server application 120 may then perform an unbind operation to terminate

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

the protocol session between the client server application 120 and the directory server 110. ...⁴⁵

A drawback currently encountered in LDAP, however, is that it requires each client to bind directly to the directory server 110 before being able to perform a request for information from the LDAP directory. ... As a result of this additional load, the directory server 110 suffers substantial degradation in performance for data retrieval.⁴⁶

The authors of the Cluet article are apparently cognizant of the degradation in performance associated with repeated binding/unbinding, and thus the Cluet article is directed to using LDAP directory caches.⁴⁷ However, the caching described in the Cluet article is performed by a client machine, and is apparently to reduce client server communications.

There has been much recent interest in the area of **client caching** to efficiently handle declarative queries using semantic information. The reasons for using the semantic information include better use of the client cache, more autonomy of the client, and reduced client server communication.⁴⁸

The Cluet article's "client" machines appear to coincide with client server applications 120 of Appellant's Figure 1 above. The Cluet article is silent as to the number of connections, and how long they are maintained, as between a caching client machine and the server.

Luotonen is directed to a proxy server caching mechanism that provides a file directory structure and mapping mechanism within the file directory structure.⁴⁹ With regard to the connections between the proxy server and remote server, however, Luotonen specifically teaches that these connections are only temporarily maintained to perform up-to-date checks.

⁴⁵ Specification Page 3, line 15 through Page 4, line 7.

⁴⁶ Specification page 4, lines 9-15.

⁴⁷ Cluet article Title.

⁴⁸ Cluet article, page 273 (emphasis added).

⁴⁹ Luotonen Title.

**Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005**

When it is desired to perform up-to-date checks, the proxy server connects to the remote server and requests that the document in question be sent, but only if the document has been modified since the most recently logged modification date/time.⁶⁰

Illustrative claim 1, by contrast, specifically recites, **"simultaneously maintaining a first plurality of connections between the directory server and a caching daemon; determining if an application is requesting information from the directory server; determining if the requested information is stored in the caching daemon in response to determining that the application has requested information; and sending the requested information to the application."** The Cluet article and Luotonen fail to teach or fairly suggest "simultaneously maintaining a first plurality of connections between the directory server and a caching daemon." The Office action dated March 7, 2005 states, "It is evident from Cluet's disclosure that he teaches maintaining a connection between the directory server and the caching daemon." This position does not follow from the teaching of the Cluet article. The Cluet article specifically teaches reducing client server communications.⁶¹ It is wholly inconsistent to allege a teaching of the Cluet article to, on the one hand, perform client caching to reduce client server communications (which reduces need for the connections), and yet on the other hand to allege that the Cluet article teaches maintaining those same connections. Appellant submits that in actuality the Cluet article teaches away from simultaneously maintaining a plurality of connections based on the teachings of reducing client server communications. If the teachings of reducing communications are discounted by the Board, then the Cluet article is at best wholly ambiguous as to the state of the connections so as not to support the position of the Office action.

As quoted above, Luotonen expressly teaches away from maintaining the connections. The Office action attempts to rely on Luotonen's Figure 1 for an

⁶⁰ Luotonen Col. 6, line 66 through Col. 7, line 4 (emphasis added).

⁶¹ Cluet article, page 273.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

alleged teaching of maintaining the connections,⁵² but the figure does not support the position of the Office action, especially since the line of the figure relied upon is dashed.⁵³ Likewise, with the teachings regarding a "proxy server cache structure that stores and accesses documents in an optimal manner....,"⁵⁴ and "the ability to locate documents in a cache without latency induced by long path names,"⁵⁵ it does not follow from these features that a plurality of connections should be maintained. Unlike the position of the Office action, however, Appellant's position that Luotonen teaches away simultaneously maintaining connections is support by the text of Luotonen.⁵⁶ Thus, the Cluet article taken with Luotonen does not teach or fairly suggest all the limitations of illustrative claim 1.

Based on the foregoing, Appellant respectfully requests that the rejections in the Office action of this group of claims be reversed, and the case set for issue.

B. Claims 21, 23-27, 29-33 and 35-44

Claims 21, 23-27, 29-33 and 35-44 stand rejected as allegedly obvious over the Cluet article in view of Luotonen. Claim 21 is representative of this group of claims. This grouping is for purposes of this appeal only, and should not be construed to mean the patentability of any of the claims may be determined, in later actions before a court, based on the grouping. Rather, the presumption of 35 U.S.C. § 282 shall apply to each claim individually.

The Cluet article is directed to using LDAP directory caches.⁵⁷ However, the caching described in the Cluet article is performed by a client machine, and is to reduce client server communications.⁵⁸ The Cluet article's "client" machines

⁵² Office action dated March 7, 2005, paragraph spanning pages 10 and 11.

⁵³ Luotonen's Figure 1 (dashed line between the proxy server 14 and the remote server 16, and note how the lines between the client 12 and proxy server 14 are not dashed).

⁵⁴ Office action dated March 7, 2005, paragraph spanning pages 10 and 11.

⁵⁵ *Id.* (emphasis added).

⁵⁶ Luotonen Col. 6, line 66 through Col. 7, line 4.

⁵⁷ Cluet Article Title.

⁵⁸ Cluet article, page 273.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

appear to coincide with client server applications 120 of Appellant's Figure 1 above. The Cluet article is silent as to the number of connections, and how long they are maintained, as between a caching client machine and the server.

Luotonen is directed to a proxy server caching mechanism that provides a file directory structure and mapping mechanism within the file directory structure.⁵⁹ The apparent advantage of Luotonen is that each long URL is converted to a fingerprint used to map the URL in the proxy server.⁶⁰ Luotonen does not appear to teach any interchangeability between the URL fingerprinting and LDAP hierarchical tree structure.⁶¹ As for connections between the proxy server and remote server, Luotonen specifically teaches that these connections are only temporarily maintained to perform up-to-date checks.⁶²

Illustrative claim 21, by contrast, specifically recites, **"simultaneously maintaining a plurality of connections between the directory server and a lightweight directory access protocol ("LDAP") caching daemon; determining if an application is requesting information from the directory server; determining if the requested information is stored in the LDAP caching daemon in response to determining that the application has requested information; and sending the requested information to the application."** The Cluet article and Luotonen fail to teach or fairly suggest "simultaneously maintaining a plurality of connections," and also fail to teach or fairly suggest a "lightweight directory access protocol ("LDAP") caching daemon."

As for "simultaneously maintaining a plurality of connections," the Office action dated March 7, 2005 states, "It is evident from Cluet's disclosure that he teaches maintaining a connection between the directory server and the caching daemon." This position does not follow from the teaching of the Cluet article. The Cluet article specifically teaches reducing client server communications.⁶³ It

⁵⁹ Luotonen Title.

⁶⁰ Luotonen Abstract.

⁶¹ With regard to the LDAP hierarchical tree structure, see Specification Page 3, lines 1-13.

⁶² Luotonen Col. 6, line 66 through Col. 7, line 4.

⁶³ Cluet article, page 273.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

is wholly inconsistent to allege a teaching of the Cluet article to, on the one hand, perform client caching to reduce client server communications (which reduces need for the connections), and yet on the other hand to allege that the Cluet article teaches maintaining those same connections. Appellant submits that in actuality the Cluet article teaches away from simultaneously maintaining a plurality of connections based on the teachings of reducing client server communications. If the teachings of reducing communications are discounted by the Board, then the Cluet article is at best wholly ambiguous as to the state of the connections so as not to support the position of the Office action.

As quoted above, Luotonen expressly teaches away from maintaining the connections. The Office action attempts to rely on Duotone's Figure 1 for an alleged teaching of maintaining the connections,⁶⁴ but figure does not support the position.⁶⁵ Likewise, with the teachings regarding a "proxy server cache structure that stores and accesses documents in an optimal manner...,"⁶⁶ and "the ability to locate documents in a cache without latency induced by long path names,"⁶⁷ it does not follow from these features that a plurality of connections should be maintained. Unlike the position of the Office action, however, Appellant's position that Luotonen teaches away from simultaneously maintaining a plurality of connections is support by the text of Luotonen.⁶⁸ For this reason alone the rejection of the Office action should be overturned.

As for a "lightweight directory access protocol ("LDAP") caching daemon," the Cluet article teaches **caching on the client machine**.⁶⁹ Luotonen teaches a **proxy server for URLs**.⁷⁰ Even if the Cluet article and Luotonen are properly

⁶⁴ Office action dated March 7, 2005, paragraph spanning pages 10 and 11.

⁶⁵ Luotonen's Figure 1 (dashed line between the proxy server 14 and the remote server 16, and note how the lines between the client 12 and proxy server 14 are not dashed).

⁶⁶ Office action dated March 7, 2005, paragraph spanning pages 10 and 11.

⁶⁷ *Id.* (emphasis added).

⁶⁸ Luotonen Col. 6, line 66 through Col. 7, line 4.

⁶⁹ Cluet article, page 273.

⁷⁰ Luotonen Title.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

considered together (which Appellant does not admit), the Cluet article and Luotonen fail to teach or suggest that the URL proxy server could or should be used as an LDAP caching daemon, that LDAP caching should be anywhere other than the client machine, or that an LDAP caching daemon should "simultaneously maintaining a plurality of connections between the directory server and" the LDAP caching daemon. For these additional reasons, the rejection of the Office action should be overturned.

Based on the foregoing, Appellant respectfully requests that the rejections in the Office action of this group of claims be reversed, and the case set for issue.

C. Non-Statutory Subject Matter Rejections⁷¹

Claims 16-20, 33, 35-38 and 42-44 stand rejected as allegedly directed to non-statutory subject matter. Claim 16 is illustrative of the claims to which the non-statutory subject matter rejections have been applied. This grouping is for purposes of this appeal only, and should not be construed to mean the patentability of any of the claims may be determined, in later actions before a court, based on the grouping. Rather, the presumption of 35 U.S.C. § 282 shall apply to each claim individually. It is noted that the Examiner issued three Office actions, including one after a request for continuing examination, and these

⁷¹ The appeal rules at 37 C.F.R. § 41.39 allows an Examiner to lodge a new ground of rejection by way of an Examiner's answer. When such occurs, the Appellants have the right to either re-open prosecution (§ 41.39(b)(1)), or elect to maintain the appeal (§ 41.39(b)(2)). In this case, however, the prosecution has been *sua sponte* re-opened by the Office, with the Office action parroting previous rejections and adding a single new rejection alleging non-statutory subject matter. The *sua sponte* re-opening has thus forced the Appellants to re-file their Notice of Appeal and Appeal brief, along with the associated fees. Appellants believe this is an attempt to circumvent both § 41.39, and in general the intent of the new rules:

Thus, the revision would improve the quality of examiner's answers and reduce pendency by providing for the inclusion of the new ground of rejection in an examiner's answer without having to reopen prosecution. By permitting examiners to include a new ground of rejection in an examiner's answer, newly presented arguments can now be addressed by a new ground of rejection in the examiner's answer when appropriate. Furthermore, if new arguments can now be addressed by the examiner by incorporating a new ground of rejection in the examiner's answer, the new arguments may be able to be addressed without reopening prosecution and thereby decreasing pendency.

69 Fed. Reg. 49959.

149158.01/1662.52300

Page 18 of 30

HP PDNO 200301685-1

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

non-statutory subject matter rejections did not appear until after Appellants submitted their first Appeal Brief on December 9, 2004.⁷²

Appellants respectfully submit that the Office action dated March 7, 2005 fails to make a *prima facie* case with respect to the non-statutory rejections.

Office personnel have the burden to establish a *prima facie* case that the claimed invention as a whole is directed solely to an abstract idea or to manipulation of abstract ideas or does not produce a useful result. **Only when the claim is devoid of any limitation to a practical application in the technological arts should it be rejected under 35 U.S.C. 101.**⁷³

In forming the alleged non-statutory rejection, the Office action dated March 7, 2005 focuses exclusively on two words from the preamble, "caching daemon." This rejection thus ignores the mandate of the MPEP to review the claimed invention as a whole.⁷⁴ Moreover, the Office action dated March 7, 2005 fails to review whether "the claim is devoid of any limitation to a practical application in the technological arts..."⁷⁵ For this reason alone, the non-statutory rejection should be reversed and these claims set for issue.

Illustrative claim 16 specifically requires, "a data cache adapted to store a subset of information from a directory server; and a controller adapted to establish and maintain a first plurality of connections to the directory server, determine if an application is requesting information from the directory server, determining if the requested information is stored in the data cache, and send the requested information to the application." Appellants respectfully submit to the Board that illustrative claim is not directed to abstract ideas; rather, the illustrative claim produces a useful result of sending requested information to the requesting

⁷² "It is essential that patent applications obtain a prompt yet complete examination of their applications. Under the principles of compact prosecution, each claim should be reviewed for compliance with every statutory requirement for patentability in the initial review.... Thus, Office personnel should state all reasons and bases for rejecting claims in the first Office action." MPEP 2106 (Eighth ed. Rev. 2, May 2004, pg. 2100-3, -4)(emphasis added).

⁷³ MPEP 2106(II)(A)(emphasis added).

⁷⁴ "[W]hen evaluating the scope of a claim, every limitation in the claim must be considered. Office personnel may not dissect a claimed invention into discrete elements and then evaluate the elements in isolation." MPEP 2106(II)(B) (emphasis original).

⁷⁵ *Id.*

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

application. For this additional reason, the non-statutory rejection should be reversed and these claims set for issue.

Finally, the timing and propriety of the non-statutory subject matter rejections are questionable given that this Examiner has found statutory and allowed claims that, in the Appellants' view, are no less statutory than illustrative claim 16.⁷⁶

Based on the foregoing, Appellants respectfully submit that claims 16-20, 33, 35-38 and 42-44 are directed to statutory subject matter, the rejection of these claims should be reversed, and the claims set for issue.

⁷⁶ A system for parsing an XML document, comprising:
a streaming parser;
a set of streaming parser APIs that interface with said streaming parser, from which set one of said streaming parser APIs is selected and instantiated by said streaming parser to parse said XML document, and including within said set
a raw streaming parser API for providing non-validating access to said XML document,
a non-validating streaming parser API for reading information for said XML document and using that information to provide access to the XML document, and
a validating streaming parser API for reading information for said XML document and using that information to validate and provide access to the XML document; and
means for selecting and instantiating one of said set of streaming parser APIs to provide access to, and parse, said XML document.

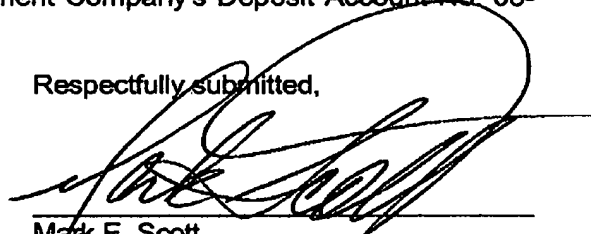
U.S. Patent No. 6,880,125, Issued April 12, 2005, and listing Rachna Singh as the assistant Examiner.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

VIII. CONCLUSION

For the reasons stated above, Appellant respectfully submits that the Examiner erred in rejecting all pending claims. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,



Mark E. Scott
PTO Reg. No. 43,100
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPELLANT

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

IX. CLAIMS APPENDIX

1. (Previously presented) A method for accessing a directory server, the method comprising:

simultaneously maintaining a first plurality of connections between the directory server and a caching daemon;
determining if an application is requesting information from the directory server;
determining if the requested information is stored in the caching daemon in response to determining that the application has requested information; and
sending the requested information to the application.

2. (Original) The method of claim 1, wherein determining if an application is requesting information from the directory server further comprises:

establishing a second connection between the application and the caching daemon; and
receiving a request for information from the application over the second connection at the caching daemon.

3. (Original) The method of claim 1, further comprising:

retrieving the requested information from the caching daemon in response to determining that the caching daemon has the information stored therein.

4. (Original) The method of claim 1, further comprising:

retrieving the requested information from the directory server in response to determining that the caching daemon has not previously stored the information.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

5. (Original) The method of claim 4, further comprising:
storing the information retrieved from the directory server at the caching daemon.
6. (Previously presented) An apparatus, comprising:
a directory server for storing information; and
a caching daemon maintaining a plurality of connections to the directory server, the caching daemon configured to determine if an application is requesting information from the directory server, determine if the requested information is stored within the caching daemon; and send the requested information to the application.
7. (Original) The apparatus of claim 6, wherein the caching daemon is further adapted to establish a second connection with the application and receive a request for information from the application over the second connection.
8. (Original) The apparatus of claim 6, wherein the caching daemon comprises:
a data cache adapted to store a subset of the information stored in the directory server; and
wherein the caching daemon is further adapted to retrieve the requested information from the data cache in response to determining that the requested information is part of the subset of information stored therein.
9. (Original) The apparatus of claim 6, wherein the caching daemon is further adapted to retrieve the requested information from the directory server in response to determining that the requested information is not stored within the data cache.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

10. (Original) The apparatus of claim 9, wherein the caching daemon is further adapted to store the requested information retrieved from the directory server in the data cache.

11. (Original) An apparatus for accessing a directory server, the apparatus comprising:

- means for establishing a first plurality of simultaneously running connections between the directory server and a caching daemon;
- means for determining if an application is requesting information from the directory server;
- means for determining if the requested information is stored in the caching daemon in response to determining that the application has requested information; and
- means for sending the requested information to the application.

12. (Original) The method of claim 11, wherein the means for determining if an application is requesting information from the directory server further comprises:

- means for establishing a second connection between the application and the caching daemon; and
- means for receiving a request for information from the application over the second connection at the caching daemon.

13. (Original) The method of claim 11, further comprising:

- means for retrieving the requested information from the caching daemon in response to determining that the caching daemon has the information stored therein.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

14. (Original) The method of claim 11, further comprising:
means for retrieving the requested information from the directory server in response to determining that the caching daemon has not previously stored the information.
15. (Original) The method of claim 14, further comprising:
means for storing the information retrieved from the directory server at the caching daemon.
16. (Previously presented) A caching daemon, comprising:
a data cache adapted to store a subset of information from a directory server; and
a controller adapted to establish and maintain a first plurality of connections to the directory server, determine if an application is requesting information from the directory server, determining if the requested information is stored in the data cache, and send the requested information to the application.
17. (Original) The caching daemon of claim 16, wherein the controller is further adapted to establish a second connection to the application and receive a request for information from the application over the second connection.
18. (Original) The caching daemon of claim 16, wherein the controller is further adapted to retrieve the requested information from the data cache in response to determining that the data cache has the information stored therein.
19. (Original) The caching daemon of claim 16, wherein the controller is further adapted to retrieve the requested information from the directory server in response to determining that the data cache does not have the information stored therein.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

20. (Original) The caching daemon of claim 19, wherein the controller is further adapted to store the information retrieved from the directory server at the data cache.

21. (Previously presented) A method for accessing a directory server, the method comprising:

simultaneously maintaining a plurality of connections between the directory server and a lightweight directory access protocol ("LDAP") caching daemon;

determining if an application is requesting information from the directory server;

determining if the requested information is stored in the LDAP caching daemon in response to determining that the application has requested information; and

sending the requested information to the application.

22. (Cancelled).

23. (Previously presented) The method of claim 21, wherein determining if an application is requesting information from the directory server further comprises:

establishing a connection between the application and the LDAP caching daemon; and

receiving a request for information from the application over the connection at the LDAP caching daemon.

24. (Previously presented) The method of claim 21, further comprising:

retrieving the requested information from the LDAP caching daemon in response to determining that the LDAP caching daemon has the information stored therein.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

25. (Previously presented) The method of claim 21, further comprising:
retrieving the requested information from the directory server in response
to determining that the LDAP caching daemon has not previously
stored the information.
26. (Previously presented) The method of claim 25, further comprising:
storing the information retrieved from the directory server at the LDAP
caching daemon.
27. (Previously presented) An apparatus, comprising:
a directory server for storing information; and
a lightweight directory access protocol ("LDAP") caching daemon, the
LDAP caching daemon configured to maintain a plurality of
connections to the directory server, determine if an application is
requesting information from the directory server, determine if the
requested information is stored within the LDAP caching daemon;
and send the requested information to the application.
28. (Cancelled).
29. (Previously presented) The apparatus of claim 27, wherein the LDAP
caching daemon is further configured to establish a connection with the
application and receive a request for information from the application over the
connection.
30. (Previously presented) The apparatus of claim 27, wherein the LDAP
caching daemon comprises:
a data cache adapted to store a subset of the information stored in the
directory server; and
wherein the LDAP caching daemon is further configured to retrieve the
requested information from the data cache in response to

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

determining that the requested information is part of the subset of information stored therein.

31. (Previously presented) The apparatus of claim 27, wherein the LDAP caching daemon is further configured to retrieve the requested information from the directory server in response to determining that the requested information is not stored within the data cache.

32. (Previously presented) The apparatus of claim 31, wherein the LDAP caching daemon is further configured to store the requested information retrieved from the directory server in the data cache.

33. (Previously presented) A lightweight directory access protocol ("LDAP") caching daemon, comprising:

- a data cache that stores a subset of information from a directory server;
- and

- a controller configured to establish and maintain a first plurality of connections to the directory server, determine if an application is requesting information from the directory server, determine if the requested information is stored in the data cache, and send the requested information to the application.

34. (Cancelled).

35. (Previously presented) The LDAP caching daemon of claim 33, wherein the controller is further configured to establish a connection to the application and receive a request for information from the application over the connection.

36. (Previously presented) The LDAP caching daemon of claim 33, wherein the controller is further configured to retrieve the requested information from the data cache if the data cache has the information stored therein.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

37. (Previously presented) The LDAP caching daemon of claim 33, wherein the controller is further configured to retrieve the requested information from the directory server if the data cache does not have the information stored therein.

38. (Previously presented) The LDAP caching daemon of claim 37, wherein the controller is further configured to store the information retrieved from the directory server at the data cache.

39. (Previously presented) A method for accessing a directory server, the method comprising:

continuously maintaining a plurality of connections between the directory server and a lightweight directory access protocol caching ("LDAP") caching daemon;

receiving from an application a request for information from the directory server;

using a connection of the plurality of connections to retrieve the requested information from the directory server; and

sending the requested information to the application.

40. (Previously presented) The method of claim 39, further comprising: retrieving the requested information from the LDAP caching daemon if the requested information is stored in the LDAP caching daemon, and wherein using a connection further comprises using the connection if the requested information is not stored in the LDAP caching daemon.

41. (Previously presented) The method of claim 39, further comprising: storing the information retrieved from the directory server at the LDAP caching daemon.

Appl. No. 09/611,920
Appeal Brief dated June 16, 2005
Reply to Office action of March 7, 2005

42. (Previously presented) A lightweight directory access protocol ("LDAP") caching daemon, comprising:

a data cache that stores a subset of information from a directory server;
and

a controller configured to establish and maintain a plurality of connections to a directory server receive from an application a request for information from the directory server, use a connection of the plurality of connections to retrieve the requested information from the directory server, and send the requested information to the application.

43. (Previously presented) The LDAP caching daemon of claim 42, wherein the controller is further configured to retrieve the requested information from the data cache if the requested information is stored in the data cache, and to use the connection if the requested information is not stored in the data cache.

44. (Previously presented) The LDAP caching daemon of claim 42, wherein the controller is further configured to store the information retrieved from the directory server at the LDAP caching daemon.